What is claimed is:

1. A processor capable of operating in any of a plurality of operating modes, the processor comprising:

a plurality w of registers, wherein fewer than w registers are program-accessible during any single mode of operation;

circuitry having n bits capable of carrying a mode value, wherein the mode value identifies a current operating mode of the processor;

circuitry having m bits capable of carrying a register value, wherein the register value identifies one of the plurality of registers;

encoder logic configured to encode the mode value with the register value to generate a mapped value carried on p bits, where p is less than m+n, wherein the mapped value collectively identifies both a register and a current processor operating mode.

2. The processor of claim 1, further comprising circuitry for passing the mapped value to different pipelined stages of the processor along with data from a register uniquely identified by the mapped value.

3. The processor of claim 1, further comprising data forwarding logic configured to compare the mapped value output from the encoder logic with at least one mapped value from a subsequent pipeline stage.

4. The processor of claim 3, wherein the data forwarding logic includes logic for reading a data value associated with the mapped value output from the encoder from a register file, responsive to the logic configured to compare, if the mapped value output from the encoder logic does not match with the at least one mapped value from a subsequent pipeline stage.

5. The processor of claim 3, wherein the data forwarding logic includes logic for reading a data value associated with the mapped value output from the encoder from a subsequent pipeline stage, responsive to the logic configured to compare, if the mapped value output from the encoder logic does match with the at least one mapped value from a subsequent pipeline stage.

6. The processor of claim 1, wherein the circuitry having n bits comprises a component selected from the group consisting of: a register, a memory device, and a latch.

7. The processor of claim 1, wherein w is 32, n is five, m is four, and p is five.

8. A processor comprising:

a plurality w of registers arranged in a banked configuration, such that fewer than w registers are program-accessible to any single instruction operation;

logic configured to carry a register-select value defined by m bits;

logic configured to carry a register-bank value defined by n bits; and

encoder logic configured to encode register-select value with the register-bank value to generate a mapped value defined by p bits output from the encoder logic, where p is less than m+n, wherein a mapped value carried on the p bits uniquely identifies a register among the plurality w of registers.

9. The processor of claim 8, further comprising circuitry for passing the mapped value to different pipelined stages of the processor along with data from a register uniquely identified by the mapped value.

10. The processor of claim 8, further comprising data forwarding logic configured to compare the mapped value output from the encoder logic with at least one mapped value from a subsequent pipeline stage.

11. The processor of claim 10, wherein the data forwarding logic includes logic for reading a data value associated with the mapped value output from the encoder from a register file, responsive to the logic configured to compare, if the mapped value output from the encoder logic does not match with the at least one mapped value from a subsequent pipeline stage.

12. The processor of claim 10, wherein the data forwarding logic includes logic for reading a data value associated with the mapped value output from the encoder from a subsequent pipeline stage, responsive to the logic configured to compare, if the mapped value output from the encoder logic does match with the at least one mapped value from a subsequent pipeline stage.

13. The processor of claim 8, wherein the logic having n bits comprises a component selected from the group consisting of: a register, a memory device, and a latch.

14. In a processor capable of operating in a plurality of modes and having w registers, wherein fewer than w registers are program accessible during the execution of a single instruction, a method comprising:

encoding a mode value carried on a plurality n of mode-identifying bits with a register value carried on a plurality m of register-select bits to produce a mapped value carried on a plurality p of bits, wherein p is less than the sum of n+m, where the mapped value carried on the p bits collectively defines both a current operating mode and register selection; and

using the encoded plurality of bits to uniquely access any of the w registers.

15. The method of claim 14, further comprising passing the mapped value to different pipelined stages of the processor along with data from a register uniquely identified by the mapped value.

16. The method of claim 14, further comprising comparing the mapped value output from the encoder logic with at least one mapped value from a subsequent pipeline stage.

17. The method of claim 16, wherein the step of comparing further comprises reading a data value associated with the mapped value, if the mapped value output from the encoding step does not match with the at least one mapped value from a subsequent pipeline stage.

18. The method of claim 16, further comprising reading a data value associated with the mapped value output from the encoder from a subsequent pipeline stage, if the mapped value output from the encoder logic does match with the at least one mapped value from a subsequent pipeline stage.

19. In a processor having w registers arranged in a banked configuration, such that fewer than w registers are accessible to any single instruction operation, a method comprising:

encoding a bank-select value carried on a plurality n of bank-identifying bit with a register value carried on a plurality m of register-select bits to produce a mapped value carried on a plurality p of bits, wherein p is less than the sum of n+m; and

using the encoded plurality of bits to uniquely access any of the w registers.

20. The method of claim 19, further comprising passing the mapped value to different pipelined stages of the processor along with data from a register uniquely identified by the mapped value.

21. The method of claim 19, further comprising comparing the mapped value output from the encoder logic with at least one mapped value from a subsequent pipeline stage.

22. The method of claim 21, wherein the step of comparing further comprises reading a data value associated with the mapped value, if the mapped value output from the encoding step does not match with the at least one mapped value from a subsequent pipeline stage.

23. The method of claim 21, further comprising reading a data value associated with the mapped value output from the encoder from a subsequent pipeline stage, if the mapped value output from the encoder logic does match with the at least one mapped value from a subsequent pipeline stage.